Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method for optimizing a netlist change order flow, wherein a design layout created by a layout tool from a reference netlist is to be changed by a modified version of the netlist, wherein both ~~netlist~~ netlists are hierarchical, the method comprising:

(a) comparing the modified netlist with the original netlist outside of the layout tool, wherein the comparing comprises generating a first instance map data structure and a first net map data structure corresponding to a flat view of the reference netlist, and generating a second instance map data structure and a second net map data structure corresponding to a flat view of the modified netlist, wherein each of the first and second instance map data structures maintain a mapping of leaf-level instance names, and wherein each of the first and second net map data structures maintain a list of nets;

(b) automatically generating at least one change order based on differences found between the ~~two netlists~~ first and second instance map data structures and between the first and second net map data structures; and

(c) applying the change order to the design layout to generate a modified design layout.

2. (Original) The method of claim 1 further including the step of:

providing a software tool for performing steps (a) and (b).

3. (Original) The method of claim 2 wherein step (a) further includes the step of: inputting the reference netlist and the modified netlist into the software tool.

4. (Original) The method of claim 3 wherein step (a) further includes the step of: comparing the reference netlist and the modified netlist in a flat manner.

5. (Original) The method of claim 4 wherein step (a) further includes the step of: creating flat views for both the reference netlist and the modified netlist.

6. (Currently amended) The method of claim 5 wherein ~~step (a) further includes the step of: generating two data structures corresponding to each of the flat views,~~ the first and second instance map data structures comprise an instance map, and wherein the first and second instance map data structures comprise a net map.

7. (Original) The method of claim 6 wherein step (a) further includes the step of:

maintaining in each of the instance maps a mapping of hierarchical leaf-level instance names and corresponding instance types, wherein modules are excluded.

8. (Original) The method of claim 5 wherein step (a) further includes the step of:

maintaining in each of the net maps a list of nets and corresponding pins for the nets across all module hierarchies.

9. (Original) The method of claim 8 wherein step (a) further includes the step of:

using a top-net is to represent a net across hierarchies and representing the pins for the net as a set.

10. (Original) The method of claim 9 wherein step (a) further includes the step of:

assuming that a name of the net in the design layout is same as the top-net name.

11. (Original) The method of claim 9 wherein step (a) further includes the step of:

comparing the flat views of the modified netlist with the flat views of the reference netlist by,

(i) sequentially reading and comparing the leaf cells in the instance maps; and

(ii) comparing the net map for the modified netlist with the net map for the reference netlist.

12. (Original) The method of claim 6 wherein step (b) further includes the step of:

generating an ECO in response to any one of the following: 1) a leaf cell is in the reference instance map, but not in the modified instance map; 2) a leaf cell is in the modified instance map, but not in the reference instance map; 3) a leaf cell is in both instance maps, but there is a cell type mismatch.

13. (Original) The method of claim 2 wherein step (c) further includes the step of:

inputting the change orders into the layout tool to apply the changes and to generate the modified layout.

14. (Currently amended) A computer-readable medium containing program instructions for optimizing a netlist change order flow, wherein a design layout created by a layout tool from a reference netlist is to be changed by a modified version of the netlist, wherein both ~~netlist~~ netlists are hierarchical, the program instructions for:

(a) comparing the modified netlist with the original netlist outside of the layout tool, wherein the comparing comprises generating a first instance map data structure and a first net map data structure corresponding to a flat view of the reference netlist, and generating a second instance map data structure and a second net map data structure corresponding to a flat view of the modified netlist, wherein each of the first and second instance map data structures maintain a mapping of leaf-level instance

names, and wherein each of the first and second net map data structures maintain a list of nets;

(b) automatically generating at least one change order based on differences found between the ~~two netlists~~ first and second instance map data structures and between the first and second net map data structures; and

(c) applying the change order to the design layout to generate a modified design layout.

15. (Original) The computer-readable medium of claim 14 further including the instruction of:

providing a software tool for performing instructions (a) and (b).

16. (Original) The computer-readable medium of claim 15 wherein instruction (a) further includes the instruction of:

inputting the reference netlist and the modified netlist into the software tool.

17. (Original) The computer-readable medium of claim 16 wherein instruction (a) further includes the instruction of:

comparing the reference netlist and the modified netlist in a flat manner.

18. (Original) The computer-readable medium of claim 17 wherein instruction (a) further includes the instruction of:

creating flat views for both the reference netlist and the modified netlist.

19. (Currently amended) The computer-readable medium of claim 18 wherein ~~step (a) further includes the step of: generating two data structures corresponding to each of the flat views,~~ the first and second instance map data structures comprise an instance map, and wherein the first and second instance map data structures comprise a net map.

20. (Original) The computer-readable medium of claim 19 wherein instruction (a) further includes the instruction of: maintaining in each of the instance maps a mapping of hierarchical leaf-level instance names and corresponding instance types, wherein modules are excluded.

21. (Original) The computer-readable medium of claim 18 wherein instruction (a) further includes the instruction of:

maintaining in each of the net maps a list of nets and corresponding pins for the nets across all module hierarchies.

22. (Original) The computer-readable medium of claim 21 wherein instruction (a) further includes the instruction of:

using a top-net is to represent a net across hierarchies and representing the pins for the net as a set.

23. (Original) The computer-readable medium of claim 22 wherein instruction (a) further includes the instruction of:

assuming that a name of the net in the design layout is same as the top-net name.

24. (Original) The computer-readable medium of claim 22 wherein instruction (a) further includes the instruction of:

comparing the flat views of the modified netlist 48 with the flat views of the reference netlist by,

(i) sequentially reading and comparing the leaf cells in the instance maps; and

(ii) comparing the net map for the modified netlist 48 with the net map for the reference netlist.

25. (Original) The computer-readable medium of claim 19 wherein instruction (b) further includes the instruction of:

generating an ECO in response to any one of the following: 1) a leaf cell is in the reference instance map, but not in the modified instance map; 2) a leaf cell is in the modified instance map, but not in the reference instance map; 3) a leaf cell is in both instance maps, but there is a cell type mismatch.

26. (Original) The computer-readable medium of claim 15 wherein instruction (c) further includes the instruction of:

inputting the change orders into the layout tool to apply the changes and to generate the modified layout.